WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Thursday, April 14, 2005

Hide?	Set Name	Query	Hit Count
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
☐	L13	L12 and l4	3
☐	L12	712/1,2,3,7,24.ccls.	835
☐	L11	L10 and @pd > 20040901	0
☐	L10	L8 and ((reduc$6 or minimiz$6) with hardware$2 with cost$2)	4
☐	L9	L8 and ((analy$6 or profil$6) with program$2)	27
☐	L8	((optimiz$6 or improv$5) with (ISA or (instruction set architecture$2)))	1840
☐	L7	L6 and ((optimiz$6 or improv$5) with (ISA or (instruction set architecture$2)))	19
☐	L6	717/$.ccls.	8274
☐	L5	(optimiz$6 or improv$5) with (ISA or (instruction set architecture$2)) with execut$5	33
☐	L4	(optimiz$8 or improv$8) with (ISA or (instruction set architecture$2)) with execut$6	33
	DB=USPT; PLUR=YES; OP=ADJ		
☐	L3	L1 and L2	2
☐	L2	717/$.ccls.	4909
☐	L1	(optimiz$8 or improv$8) with (ISA or (instruction set architecture$2)) with execut$6	15

END OF SEARCH HISTORY

Results for **"(generate<in>ab) <and> (instruction set architecture<in>ab) <and> (extension<in>ab)"**
Your search matched **2** of **1144303** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

e-mail

» View Session History

» New Search

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

Modify Search

(generate<in>ab) <and> (instruction set architecture<in>ab) <and> (extension<

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

Select **Article Information**

☐ **1. Memory traffic and data cache behavior of an MPEG-2 software decoder**
Soderquist, P.; Leeser, M.;
Computer Design: VLSI in Computers and Processors, 1997. ICCD '97. Proceedings., International Conference on
12-15 Oct. 1997 Page(s):417 - 422
AbstractPlus | Full Text: PDF(732 KB) **IEEE CNF**

☐ **2. Automatic test generation for micro-architectural verification of configurable mic cores with user extensions**
Bhattacharyya, N.; Wang, A.;
High-Level Design Validation and Test Workshop, 2001. Proceedings. Sixth IEEE Inter
7-9 Nov. 2001 Page(s):14 - 15
AbstractPlus | Full Text: PDF(35 KB) **IEEE CNF**

☐ Search Results

Results for "(extension <in>metadata) <and> (instruction set architecture<in>metadata) <and> (generate<in>metadata)"

e-mail

Your search matched 3 of 1144315 documents.

A maximum of 100 results are displayed, 25 to a page, sorted by Relevance in Descending order.

» View Session History

» New Search

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

Modify Search

(extension <in>metadata) <and> (instruction set architecture<in>metadata) <an

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

Select Article Information

☐ 1. **Memory traffic and data cache behavior of an MPEG-2 software decoder**
Soderquist, P.; Leeser, M.;
Computer Design: VLSI in Computers and Processors, 1997. ICCD '97. Proceedings., International Conference on
12-15 Oct. 1997 Page(s):417 - 422
AbstractPlus | Full Text: PDF(732 KB) IEEE CNF

☐ 2. **Hardware/software co-design of a fuzzy RISC processor**
Salapura, V.; Gschwind, M.;
Design, Automation and Test in Europe, 1998., Proceedings
23-26 Feb. 1998 Page(s):875 - 882
AbstractPlus | Full Text: PDF(236 KB) IEEE CNF

☐ 3. **Automatic test generation for micro-architectural verification of configurable mic cores with user extensions**
Bhattacharyya, N.; Wang, A.;
High-Level Design Validation and Test Workshop, 2001. Proceedings. Sixth IEEE Inter
7-9 Nov. 2001 Page(s):14 - 15
AbstractPlus | Full Text: PDF(35 KB) IEEE CNF

IEEE Xplore RELEASE 2.0

Results for "(extension <in>metadata) <and> (instruction set architecture<in>metadata) <and> (generate<in>metadata)"

e-mail

Your search matched **3** of **1144315** documents.

A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» View Session History

» New Search

Modify Search

(extension <in>metadata) <and> (instruction set architecture<in>metadata) <an

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

Select **Article Information**

☐ **1. Memory traffic and data cache behavior of an MPEG-2 software decoder**
Soderquist, P.; Leeser, M.;
Computer Design: VLSI in Computers and Processors, 1997. ICCD '97. Proceedings., International Conference on
12-15 Oct. 1997 Page(s):417 - 422
AbstractPlus | Full Text: PDF(732 KB) **IEEE CNF**

☐ **2. Hardware/software co-design of a fuzzy RISC processor**
Salapura, V.; Gschwind, M.;
Design, Automation and Test in Europe, 1998., Proceedings
23-26 Feb. 1998 Page(s):875 - 882
AbstractPlus | Full Text: PDF(236 KB) **IEEE CNF**

☐ **3. Automatic test generation for micro-architectural verification of configurable mic cores with user extensions**
Bhattacharyya, N.; Wang, A.;
High-Level Design Validation and Test Workshop, 2001. Proceedings. Sixth IEEE Inter
7-9 Nov. 2001 Page(s):14 - 15
AbstractPlus | Full Text: PDF(35 KB) **IEEE CNF**

Terms used
generate instruction set architecture extension optimize

Found **5** of **153,034**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 5 of 5

Relevance scale

1 Hardware/software co-design of a fuzzy RISC processor
V. Salapura, M. Gschwind
February 1998 **Proceedings of the conference on Design, automation and test in Europe**
Full text available: pdf(103.33 KB) Publisher Site
Additional Information: full citation, abstract, references, citings, index terms

In this paper, we show how hardware/software co-evaluation can be applied to instruction set definition. As a case study, we show the definition and evaluation of instruction set extensions for fuzzy processing. These instructions are based on the use of subword parallelism to fully exploit the processor's resources by processing multiple data streams in parallel. The proposed instructions are evaluated in software and hardware to gain a balanced view of the costs and benefits of each instructio ...

Keywords: hardware/software co-evaluation, hardware/software co-design, processor core, MIPS RISC processor, fuzzy processing, fuzzy rule evaluation, application specific instruction set processor (ASIP), subword parallelism, VHDL, logic synthesis, instruction set definition, instruction set architecture, performance evaluation

2 Fast Prolog with an extended general purpose architecture
Bruce K. Holmer, Barton Sano, Michael Carlton, Peter Van Roy, Ralph Haygood, William R. Bush, Alvin M. Despain, Joan M. Pendleton, Tep Dobry
May 1990 **ACM SIGARCH Computer Architecture News , Proceedings of the 17th annual international symposium on Computer Architecture**, Volume 18 Issue 3
Full text available: pdf(1.19 MB) Additional Information: full citation, abstract, references, citings, index terms

Most Prolog machines have been based on specialized architectures. Our goal is to start with a general purpose architecture and determine a minimal set of extensions for high performance Prolog execution. We have developed both the architecture and optimizing compiler simultaneously, drawing on results of previous implementations. We find that most Prolog specific operations can be done satisfactorily in software; however, there is a crucial set of features that the architecture must suppor ...

3 The PL/EXUS language and virtual machine
Gary A. Sitton, Thomas A. Kendrick, A. Gil Carrick
November 1973 **ACM SIGPLAN Notices , Proceedings of a symposium on High-level-language computer architecture**, Volume 8 Issue 11

Full text available: pdf(640.09 KB) Additional Information: full citation, abstract, references

This paper describes a high level general purpose language which evolved from another high level systems programming language. As well, the compiler, pseudocode, and virtual machine are discussed in some detail. The new language is a powerful PL/1 dialect, as is its parent language, XPL[1]. PL/EXUS (<u>P</u>rogramming <u>L</u>anguage/<u>E</u>xtended <u>X</u>PL <u>U</u>sers' <u>S</u>uperset), was created to satisfy a ...

4 The PL/EXUS language and virtual machine

Gary A. Sitton, Thomas A. Kendrick, A. Gil Carrick

November 1973 **Proceedings of the ACM-IEEE symposium on High-level-language computer architecture**

Full text available: pdf(650.16 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes a high level general purpose language which evolved from another high level systems programming language. As well, the compiler, pseudocode, and virtual machine are discussed in some detail. The new language is a powerful PL/1 dialect, as is its parent language, XPL 1. PL/EXUS (Programming Language/Extended XPL Users' S

5 High performance integrated Prolog processor IPP

S. Abe, T. Bandoh, S. Yamaguchi, K. Kurosawa, K. Kiriyama

June 1987 **Proceedings of the 14th annual international symposium on Computer architecture**

Full text available: pdf(652.88 KB) Additional Information: full citation, abstract, references, citings, index terms

To realize the highest performance possible for a sequential processor, and to realize utilization of a large amount of existing software, an integrated Prolog processor (IPP) and its optimized compiler are now being developed. A tagged architecture under constraints of a general purpose computer and a memory management strategy to achieve a high performance are discussed and then an IPP architecture is presented. Based on the Prolog instruction set, which is an extension of Warr ...

Results 1 - 5 of 5

Terms used
generate instruction set architecture extension optimize

Found **5** of **153,034**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 5 of 5

Relevance scale

1 Hardware/software co-design of a fuzzy RISC processor
V. Salapura, M. Gschwind
February 1998 **Proceedings of the conference on Design, automation and test in Europe**
Full text available: pdf(103.33 KB) Publisher Site
Additional Information: full citation, abstract, references, citings, index terms

In this paper, we show how hardware/software co-evaluation can be applied to instruction set definition. As a case study, we show the definition and evaluation of instruction set extensions for fuzzy processing. These instructions are based on the use of subword parallelism to fully exploit the processor's resources by processing multiple data streams in parallel. The proposed instructions are evaluated in software and hardware to gain a balanced view of the costs and benefits of each instructio ...

Keywords: hardware/software co-evaluation, hardware/software co-design, processor core, MIPS RISC processor, fuzzy processing, fuzzy rule evaluation, application specific instruction set processor (ASIP), subword parallelism, VHDL, logic synthesis, instruction set definition, instruction set architecture, performance evaluation

2 Fast Prolog with an extended general purpose architecture
Bruce K. Holmer, Barton Sano, Michael Carlton, Peter Van Roy, Ralph Haygood, William R. Bush, Alvin M. Despain, Joan M. Pendleton, Tep Dobry
May 1990 **ACM SIGARCH Computer Architecture News , Proceedings of the 17th annual international symposium on Computer Architecture**, Volume 18 Issue 3
Full text available: pdf(1.19 MB) Additional Information: full citation, abstract, references, citings, index terms

Most Prolog machines have been based on specialized architectures. Our goal is to start with a general purpose architecture and determine a minimal set of extensions for high performance Prolog execution. We have developed both the architecture and optimizing compiler simultaneously, drawing on results of previous implementations. We find that most Prolog specific operations can be done satisfactorily in software; however, there is a crucial set of features that the architecture must suppor ...

3 The PL/EXUS language and virtual machine
Gary A. Sitton, Thomas A. Kendrick, A. Gil Carrick
November 1973 **ACM SIGPLAN Notices , Proceedings of a symposium on High-level-language computer architecture**, Volume 8 Issue 11

Full text available: pdf(640.09 KB) Additional Information: full citation, abstract, references

This paper describes a high level general purpose language which evolved from another high level systems programming language. As well, the compiler, pseudocode, and virtual machine are discussed in some detail. The new language is a powerful PL/1 dialect, as is its parent language, XPL[1]. PL/EXUS (<u>P</u>rogramming <u>L</u>anguage/<u>E</u>xtended <u>X</u>PL <u>U</u>sers' <u>S</u>uperset), was created to satisfy a ...

4 The PL/EXUS language and virtual machine

Gary A. Sitton, Thomas A. Kendrick, A. Gil Carrick

November 1973 **Proceedings of the ACM-IEEE symposium on High-level-language computer architecture**

Full text available: pdf(650.16 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes a high level general purpose language which evolved from another high level systems programming language. As well, the compiler, pseudocode, and virtual machine are discussed in some detail. The new language is a powerful PL/1 dialect, as is its parent language, XPL 1. PL/EXUS (Programming Language/Extended XPL Users' S

5 High performance integrated Prolog processor IPP

S. Abe, T. Bandoh, S. Yamaguchi, K. Kurosawa, K. Kiriyama

June 1987 **Proceedings of the 14th annual international symposium on Computer architecture**

Full text available: pdf(652.88 KB) Additional Information: full citation, abstract, references, citings, index terms

To realize the highest performance possible for a sequential processor, and to realize utilization of a large amount of existing software, an integrated Prolog processor (IPP) and its optimized compiler are now being developed. A tagged architecture under constraints of a general purpose computer and a memory management strategy to achieve a high performance are discussed and then an IPP architecture is presented. Based on the Prolog instruction set, which is an extension of Warr ...

Results 1 - 5 of 5